Minimizing Response Time of IoT-Based Traffic Information System through Decentralized Server System

ABSTRACT

Many metropolises seek to relieve traffic congestions and reduce vehicle accidents by implementing Intelligent Traffic Information Systems. These systems manage continuous communication between vehicles, various roadside IoT devices and a central server in real time for traffic control and vehicle navigations. Short response time is critical to the success of these time-sensitive systems. For a small area, a system with centralized server architecture may just work fine. For a larger area with more IoT devices and traffic, however, the system may experience excessive response time as a result of increased network distance and constrained server processing capacity. We propose a decentralized server system to properly manage and reduce service response time. We have also developed a binary nonlinear constrained programming model with Genetic Algorithm for a heuristic solution.

1. Introduction

Many metropolises seek to reduce traffic congestions, vehicle accidents, and pollutions by implementing Intelligent Traffic Information Systems [2]. These systems manage continuous communication between vehicles, traffic control systems and various roadside Internet of Things (IoT) devices with sensors and processing servers. It measures the real-time traffic density, weather condition and controls the traffic congestion on road through dynamic management of traffic signals and direction and guidance for traveling vehicles.

Vehicles are increasingly becoming connected and are ready to interact with IoT devices in real time by sending and receiving data continuously. Such an infrastructure are both supported by private industry and by government agencies as well (https://www.its.dot.gov/cv_basics/index.htm). Data collected by these IoT devices are then fed to a central server in real time, which, in turn, performs analysis and gives instructions back to the IoT devices. The IoT devices will then relay back to traffic control systems and vehicles to help with traffic controls.

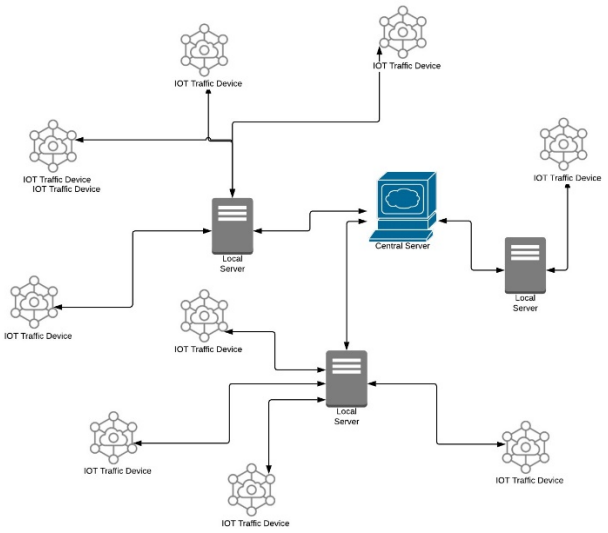
The term the Internet of Things (IoT) was coined by Kevin Ashton of Procter & Gamble in 1999 [9]. IoT has since then received significant attention both in academia and industry during the past decade. It prescribes a world where numerous smart objects are connected to each other with no human intervention. IoT has been used in many smart applications for healthcare, home and office, agriculture, equity trading [18], etc. In transportation, various IoT sensors are available and many are currently deployed to help control, manage the traffic information systems efficiently.

In general, an intelligent traffic information system needs to offer fast services to keep up with fluid, sometimes chaotic, and continuous traffic. The success of these time-sensitive systems is partially determined by their service response time. For a small area, a centralized server architecture may work just fine. For a larger area with more IoT devices and high volume of traffic, however, the system may experience excessive response time as a result of increased network distance and constrained server processing capacity. Properly managing and reducing response time is a critical requirement for traffic information systems to achieve their goals.

An alternative solution is to deploy a decentralized traffic information system. There can be three major players: a central server, multiple local servers, and numerous IoT devices. In this infrastructure, vehicles communicate directly with IoT devices nearby in real time, report key vital statistics, including speed and vehicles types, and request services for traffic guidance. IoT devices then relay this information directly to local servers nearby for speedy processing. Local servers, then process the information and give guidance back the vehicles through the IoT devices they interact with. At the same time, the local servers also serve as intermediaries between IoT devices and the central server. They forward important local traffic information to the central server. The central server, in turn, process the information at an aggregated level and communicate back to local server for global traffic directions. In essence, the central server is responsible for managing all the communications with IoT devices through intermediary local servers and overall traffics in the metropolitan (figure 1).

Figure 1. Decentralized Traffic Information Server Systems

URI: https://hdl.handle.net/10125/64386
978-0-9981331-3-3



The performance of a time-sensitive decentralized service is largely measured by its response time. Response time includes local processing time and network response time. Network response time is largely determined by network latency. Network latency refers to the amount of time that a packet of data takes to travel from one location to another on a network [8]. Minimizing service response time, as a result, requires reducing local processing time and decreasing the network latencies between servers and clients. A local server handles much of the request of IoT devices in real time and only need to connect with the central server for global traffic management. Network latency is closely related to the physical proximity between IoT devices and their assigned local servers. Instead of connecting IoT devices to a distant central server, we can locate many local servers physically near them for service request to reduce overall network latency.

The strategic placement of the local servers on a network, therefore, becomes critical in improving network latency and service response time. Since there will be many communications between local servers and the central server for global traffic management. The distance between them will also need to be reduced by optimal locating the central server on the same network. To minimize local server processing time, we can choose more capable server equipment and software package within a budget.

The main purpose of this research is to provide a framework that can guide a metropolitan to locate and manage its local and central servers to improve traffic services. We developed a binary nonlinear constrained programming model with budget and service response time constraints. This is a NP-hard combinatorial optimization problem. We propose to use Genetic Algorithm to solve the problem. Genetic Algorithm, a widely used and proven metaheuristic method for solving the problem of NP-hard and NP-complete complexities, is particularly applicable for a Stochastic Nonlinear Constrained Optimizing Problem. The proposed model are solved using the MATLAB R2017b Genetic Algorithm solver.

2. Literature review

IoT devices are widely used in smart cities and in particular, managing traffics [3, 11, 12, 17]. Andreas etc. [2] suggested that to manage and control traffic flows, the IoT devices need to capture the conditions of the road traffic with speed, flow, and density on a specific segment of the road. They proposed a framework to utilize the various traffic management sources efficiently in the context of traffic management and analyzed how different types of traffic models and algorithms can use the data sources and key functionalities of active traffic management such as short-term prediction and control. Rath [13] argued that the growth of population and vehicles causes traveling delays and contributes to environmental pollution and therefore recommend a smart IoT based system to alleviate the problem. Al-Sakran [1] proposed an intelligent traffic administration system, based on IoT, which features low cost, high scalability, high compatibility, easy to upgrade, to replace traditional traffic management system to improve road traffic tremendously.

Avasalcai etc. [4] suggested that for real-time applications with fast response times requirement, fog [5] and edge computing [16] will be the key infrastructures for deployment. Both methods locate computing resources closer to IoT devices. Raptis etc. [14] argued that the distribution of data generated by IoT technologies needs to be improved continuously. A centralized system with data being transferred back and forth in the network may lead to severely sub-optimal paths and communication overhead and ultimately increase overall network latency. To solve the problem, they proposed an edge data distribution system where services are distributed to nodes near IoT devices.

In particular, for IoT-Based Traffic Information System to work efficiently, network latency needs to be carefully managed and reduced if possible. Traffic IoT sensors are implemented on a distributed network. Service requests from IoT devices generate many messages to discover, negotiate, and invoke these services for traffic management. In addition to technical consideration, managerial issues are also important factors to the success of system. All cities face budget and procurement constraints and need to work with them judiciously. In this study, we propose a model to minimize overall response time by optimally locating

local/intermediary servers and a central server with budget constraint and maximum response time constraint to serve all IoT devices connected on the network.

3. Decision model

For convenience, we assume there is a network where we can locate IoT devices, local servers, and one central server. We assume that J number of IoT devices have already been deployed and each will generate a demand for service D_j. Given the fluidity of the traffic condition, we assume D_j is stochastic. There will be one central server and M different type of local servers we can purchase at price P_m with service capacity CP_m. We assume servers with higher capacity will command higher price. On the same network, there are I possible locations for local servers and K possible locations for the central server location. The distance between local server I and center server k is f_{ik} and the distance between local server location i and IoT device j is d_{ij}. The fixed cost of locating a local server on location i is FI_i and the fixed cost of locating the central server on location k is FS_k. We further assume that the maximum tolerable response time for service is T and the total budget is B.

Table1. Summary of Notation

Parameters

- M: number of local server types; m=1 … M
- J: total number of IoT devices; j= 1… J
- I: possible locations for local servers; i= 1…I
- K: possible locations for central server location; k= 1…K
- D_j: demand from each IoT device j (stochastic)
- CP_m: Capacity (total number of demands that can be serviced) of local server type m
- d_{ij}: distance between local server location i and IoT device j
- f_{ik}: distance between local server location i and central server location k
- FI_i: fixed cost of locating a local server on location i
- FS_k: fixed cost of locating central server on location k; ($FI_i < FS_k$)
- P_m: Price of local server type m
- P_c: Price of central server
- t: time to receive data per unit of distance
- T: maximum tolerable response time (if the response time exceeds T, it leads to time out)
- B: total available budget

Decision Variables

- X_{mi}: binary variable; takes 1 if local server m is located on location i
- Y_k: binary variable; takes 1 if central server is located on location k
- Z_{ij}: binary variable; takes 1 if IoT j gets service from local server located on i_{th} location

First, the deterministic version of the model is formulated as:

$$\text{Min P1} = \sum_{j=1}^{J}\sum_{i=1}^{I} d_{ij}Z_{ij} + \sum_{i=1}^{I}\sum_{k=1}^{I} f_{ik}Y_kX_{mi}$$

St:

$$X_{mi} - Z_{ij} \geq 0 \quad ; \quad \forall i \tag{1}$$

$$\sum_{j=1}^{J} D_jZ_{ij} \leq CP_mX_{mi} \quad ; \quad \forall i \,\&\, m \tag{2}$$

$$\sum_{i=1}^{I} Z_{ij} = 1 \quad ; \quad \forall j \tag{3}$$

$$\sum_{m=1}^{M} X_{mi} \leq 1 \quad ; \quad \forall i \tag{4}$$

$$\sum_{k=1}^{K} Y_k = 1 \quad ; \quad \forall k \tag{5}$$

$$t.d_{ij}Z_{ij} \leq T \quad ; \quad \forall i,j \tag{6}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{I} P_mX_{mi} + \sum_{i=1}^{I}\sum_{m=1}^{M} FI_iX_{mi} + \sum_{k=1}^{I} FS_kY_k + \text{Pc} \leq B \tag{7}$$

$$X_{mi}, Z_{ij}, Y_k \in \{0,1\}, \ \forall m,i,j,k$$

The objective function P1 is designed to minimize the overall response time of the entire system. As the response time is a linear function of distance so P1 is formulated to minimize the total distance between IoT devices and local servers and distance between local servers and the central server. The proposed model is to optimize the location allocation problem subject to seven constraints.

Constraint (1) is to ensure that we only assign IoT device j to location i if a local server is to be deployed on location i.

Constraint (2) is to ensure that total IoT demand for connecting to each local server doesn't exceed the local server's capacity. Capacity is determined by the types of local server deployed.

Constraint (3) is to ensure that an IoT device should be assigned to one local server to respond the demand.

Constraint (4) is to ensure that on each possible location, maximum one local server can be located.

Constraint (5) is to ensure that only one central server should be located.

Constraint (6) is designed for preventing time out in service. It helps to ensure minimum service levels and the service time doesn't exceed the maximum tolerable time.

Constraint (7) is formulated for satisfying the budget limitation.

4. Chance constrained programming

In CCP, the objective function should be achieved with the stochastic constraints held at least α of time, where α is provided as an appropriate safety margin by the decision maker [6].

Assume that x is a decision vector, ξ is a stochastic vector, and gj(x, ξ) are stochastic constraint functions, j= 1, 2… p. Since the stochastic constraints gj(x, ξ) ≤ 0, j= 1, 2… p does not define a deterministic feasible set, they need to be held with a confidence level α. Thus chance constraint is represented as follows [10]:

Pr { gj(x, ξ) ≤ 0, j= 1, 2, …, p } ≥ α (8)

Which is considered the same α for all stochastic constraints, and when we want to assume that they are different, it can be shown as follows:

Pr { gj(x, ξ) ≤ 0} ≥ αj , j= 1, 2, …, p (9)

Theorem (1): Assume that the stochastic vector $\zeta = (a_1, a_2, \dots, a_n, b)$ and the function g(x, ξ) has the form $g(x, \xi) = a_1x_1 + a_2x_2 + \cdots + a_nx_n - b$. If a_i and b are assumed to be independently normally distributed random variables, then Pr { g(x, ξ) ≤ 0} ≥ α if and only if

$$\sum_{i=1}^{n} E[a_i]x_i + \Phi^{-1}(\alpha)\sqrt{\sum_{i=1}^{n} Var[a_i]x_i^2 + V[b]} \le E[b] \quad (10)$$

Where Φ is the standardized normal distribution function. The proof of the above theorem is in [12].

In this paper, we assume Dj, potential demand from each IoT device j, is stochastic and it follows normal distribution so its notation will be changed to a random variable as $\widetilde{D_j}$. In the proposed model, constraint (2) is the only constraint that includes stochastic parameter $\widetilde{D_j}$ so using equation (10), it is turned to chance constraint as following:

$$\sum_{j=1}^{J} E\left[\widetilde{D_j}\right]Z_{ij} + \sum_{j=1}^{J} \Phi^{-1}(\alpha)\sqrt{Var\left[\widetilde{D_j}\right]Z_{ij}^{\,2}} - CP_mX_{mi} \le 0 \quad ; \quad \forall\, i, m \quad (11)$$

5. Overall Approach and Methodology

This proposed model is a Binary Constrained NLP where it includes one nonlinear constraint and objective as well. The model includes a collection of constraints: equality, inequality, linear and nonlinear constraints. This is a NP-hard combinatorial optimization problem.

In other words, optimal solutions can be obtained within a reasonable amount of time only for small-sized problems. However, problems of large size need heuristics and also metaheuristics that take advantage of the structures of the problem. In this research, Genetic Algorithm (GA) as a popular valid and appropriate metaheuristic method for solving the problem in NP-hard and NP-complete complexities level, is used for optimizing developed stochastic Knapsack Problem. GA is an evolutionary algorithm developed originally by Holland [7]. GA, based on the mechanism of genetics and natural selection, is capable of efficiently locating near optimal or even optimal solutions for many combinatorial optimization problems.

We employed MATLAB to solve the problem. MATLAB mainly works with two approaches: Problem based approach and solver-based approach. Based on the features and limitations of our problem, we chose the first one. Then, we will conduct numerical experiments to demonstrate the agility and robustness of the model.

6. Numerical Example

The following hypothetical numerical example is considered to demonstrate the agility and robustness of the proposed model.

For the experiment, we created a network of communication to locate IoT devices, local servers and central servers. The locations of IoT devices are known. Each of the IoT device generates a number of requests per unit of time with a normal distribution with a mean of 100 and standard distribution of 20 in order to address the demand uncertainty. As mentioned in previews section, we deployed Chance Constrained Programming to handle uncertain parameters embedded in model, so we needed to set the confidence level (α) to reflect the level of satisfaction for chance constraint (11). In this example we set the α as 0.9.

We limit the number of deployed IoT devices to 200. There are 20 possible locations for the local servers. Each location can host one of the three different types of local servers. The three different type of local server each command a fixed cost of $10,000, $20,000 and $30,000 respectively. They also offer different capacity of handling 10,000, 30,000, and 50,000 requests respectively. The fixed cost of locating a local server on an available location is randomly generated in a range between $1,000 and $5,000.

There are 10 possible locations for the central server. The fixed cost of locating a central server on an available location is randomly generated in a range between $10,000 and $50,000. The price of the central server is set at $1,000,000.

The distance between local server location and IoT device and distance between local server location and central server location are all randomly generated in a range between 100 and 5000 feet.

The time to transmit data per mile is assumed to be 8.2 microseconds [15]. The maximum tolerable response time is set at 3 microseconds. The overall budget is $5 million.

We ran the data using GA algorithm and were able to obtain optimal solutions with all IoT devices serviced within the tolerance of the time. The results including some of the decision variables and the optimized objective function are represented in Table 2.

TABLE 2. RESULTS INCLUDING DECISION VARIABLES AND OBJECTIVE FUNCTION

Objective Function	96,958,431 microseconds
Number of Local Servers Deployed	9 Typ1: 8 Type2:0 Type3:1

7. Conclusions

In this paper, we proposed a decentralized server system to properly manage and reduce response time in traffic information systems. In such a system, multiple local servers can be strategically located in different areas throughout the entire metropolis. These local servers collect and process data from nearby IoT devices and give speedy feedbacks for guidance. At the same time, the local servers also serve as intermediaries to communicate with a central server for overall traffic controls. We developed a binary nonlinear constrained programming model. This is a NP-hard combinatorial optimization problem. We used Genetic Algorithm to solve the problem. In the future, we will then follow up with multiple sensitivity analysis on factors including stochastic constraint satisfaction, demand, and capacity. This will help us with managerial implications of the model and help cities better allocate resources to meet the traffic demand.

8. References

[1] H.O. Al-Sakran, "Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology", International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015 37.

[2] Allström, A., J. Barcelo, J. Ekström, E. Grumert, D. Gundlegård, and C. Rydergren, Traffic Management for Smart Cities, Part of: Designing, developing, and facilitating smart cities: urban design to IoT solutions, Angelakis, V., Tragos, E., Pöhls, H.C., Kapovits, A., Bassi, A. (Eds.) (eds), 2016, pp. 211-240.

[3] S. An, B. Lee, and D. Shin, "A Survey of Intelligent Transportation Systems", Third International Conference on Third International Conference on Computational Intelligence, Communication Systems and Networks, (2011).

[4] C. Avasalcai, and D. Schahram, "Latency-aware decentralized resource management for IoT applications", In Proceedings of the 8th International Conference on the Internet of Things (IOT '18). ACM, New York, NY, USA, Article 30, 2018.

[5] B. Flavio, and R. Milito, "Fog Computing and its Role in the Internet of Things", 1st ACM Mobile Cloud Computing Workshop (2012), pp. 13–15.

[6] K. Hassanlou, "A Multi Period Portfolio Selection Using Chance Constrained Programming", Decision Science Letter, 6, 2016, 221–232.

[7] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, 1975.

[8] J.M Johansson,"On the Impact of Network Latency on Distributed Systems Design", Information Technology Management, (1), 2000, 183-194.

[9] A. Kevin, "That 'Internet of Things' Thing", RFID Journal, 22 June 2009.

[10] Liu, B., Theory and Practice of Uncertain Programming, 3rd ed., UTLAB, 2009.

[11] F.C. Nemtanu, C. Dumitrescu, C.V. Banu, and G.S. Banu, "Monitoring System with Applications in Road Transport", 20th International Symposium for Design and Technology in Electronic Packaging (SIITME) (2014).

[12] A. Ramazani, and H. Vahdat-Nejad, "A New Context-Aware Approach to Traffic Congestion", 4th International Conference on Computer and Knowledge Engineering (ICCKE) (2014).

[13] M. Rath, "Smart Traffic Management System for Traffic Control using Automated Mechanical and Electronic Devices", 2018, IOP Conference Series: Materials Science and Engineering.

[14] T.P. Raptis, A. Passarella and M. Conti, "Maximizing Industrial IoT Network Lifetime under Latency Constraints Through Edge Data Distribution", 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, (2018), pp. 708-713.

[15] F. Sherman, "Network Latency Milliseconds Per Mile", https://www.techwalla.com/articles/network-latency-milliseconds-per-mile

[16] W. Shi and S. Dustdar. "The Promise of Edge Computing", Computer 49, 5 (May 2016), pp. 78–81.

[17] S. Sukode, S. Gite, and H. Agrawal, "Context Aware Framework in IoT: A survey", International Journal of Advanced Trends in Computer Science and Engineering, vol 4, no. 1. (2015).

[18] Y. Sun, and M. Hassanlou, "Equity Trading Server Allocation Using Chance Constrained Programming", Journal of Supply Chain and Operations Management, Vol 17, Issue 1 (2019).